Multilevel Noise Contrastive Network for Few-Shot Image Denoising

Bo Jiang, Jiahuan Wang, Yao Lu, Guangming Lu, *Senior Member, IEEE*, and David Zhang, *Life Fellow, IEEE*

***Abstract*— In recent years, most denoising methods based on deep convolutional neural networks heavily rely on massive noisy–clean image pairs. Collecting massive noisy–clean image pairs is expensive and not practical in real scenes. Currently, few-shot learning has been applied to many areas to cope with the absence of data. The few-shot learning, however, in image denoising severely suffers from domain gap problems, including dataset domain gap and feature domain gap, especially for the real noisy images. Therefore, this article proposes a multilevel noise contrastive network (MNC-Net) performing few-shot image denoising. MNC-Net consists of two training stages: 1) using contrastive learning to self-supervise the training of multilevel noise contrastive learner (MNCL) on the pure synthetic noisy images with multiple Gaussian noise levels to ease the acute dataset domain gap and 2) features generated by the MNCL on limited data are fused to the second stage and alleviate the feature domain gap using our proposed denoising network. Specifically, the MNCL consists of a contrastive feature extractor (CFE) and a contrastive feature projector (CFP). MNCL learns the rich and complex content-invariant degradations and general multiple-level noise representations. The denoising network in the second stage is composed of guided feature encoder (GFE) and adaptive denoising decoder (ADD). The GFE uses contrast features from CFE to guide the produced representations on the specific input noisy images. Then, such output features are fed into the ADD to adaptively denoise the noisy images on the corresponding noise distribution. To the best of our knowledge, this work is the first attempt to jointly use the few-shot learning and contrastive learning in the deep denoising field. Extensive experiments on CBSD68, Kodak24, Set12, SIDD, and DND show that our method achieves promising denoising performances in the absence of data.**

***Index Terms*— Contrastive learning, deep learning, few-shot learning, image denoising, multilevel noise.**

Manuscript received 10 March 2022; revised 18 June 2022; accepted 29 June 2022. Date of publication 9 August 2022; date of current version 1 September 2022. This work was supported in part by the Guangdong Shenzhen joint Youth Fund under Grant 2021A151511074, in part by the NSFC Fund under Grant 62176077, in part by the Guangdong Basic and Applied Basic Research Foundation under Grant 2019B1515120055, in part by the Shenzhen Key Technical Project under Grant 2020N046, in part by the Shenzhen Fundamental Research Fund under Grant JCYJ20210324132210025, and in part by the Medical Biometrics Perception and Analysis Engineering Laboratory, Shenzhen, China. The Associate Editor coordinating the review process was Dr. Yu Yang. *(Bo Jiang and Jiahuan Wang contributed equally to this work.) (Corresponding authors: Yao Lu; Guangming Lu.)*

Bo Jiang, Jiahuan Wang, Yao Lu, and Guangming Lu are with the Department of Computer Science and Technology, Harbin Institute of Technology at Shenzhen, Shenzhen 518057, China (e-mail: jiangbo_phd@outlook.com; wangjiahuanszhit@163.com; luyao2021@hit.edu.cn; luguangm@hit.edu.cn).

David Zhang is with the School of Science and Engineering, The Chinese University of Hong Kong at Shenzhen, Shenzhen 518172, China (e-mail: davidzhang@cuhk.edu.cn).

Digital Object Identifier 10.1109/TIM.2022.3189739

I. Introduction

CURRENTLY, some denoising methods based on deep convolutional neural networks (CNNs) have achieved excellent performances due to the design of network structures or training strategies. These methods [1]–[5] heavily rely on considerable noisy–clean image pairs. However, it is difficult and expensive to collect such considerable noisy–clean image pairs, especially for the real-world noisy–clean image pairs in practice. Moreover, training the network with lots of noisy–clean image pairs also leads to high training time costs and consumes a lot of power. It severely limits the application of deep denoising methods in many practical applications. In contrast, humans can learn new visual perceptions from only a few examples [6], [7]. They can also easily perceive new visual examples. To address these problems, we want to introduce the few-shot learning into the image denoising area. Few-shot learning is usually employed in image classification, which provides satisfactory learning capability from only a few examples. Because denoising images are completely different from the existing application areas using few-shot learning, current popular few-shot frameworks [8]–[11] cannot be directly applied to the few-shot image denoising problem. Therefore, the few-shot learning used in the image denoising area is still under exploration.

Meanwhile, the strategy for the few-shot task (such as image classification and image segmentation) usually involves pretraining on a base dataset containing a large amount of data [7], [12] and fine-tuning on a novel dataset containing scarce data. Unfortunately, it is difficult to produce a positive effect when the distribution of the base dataset largely differs from that of the novel dataset. This is mainly because the network parameters trained on massive base datasets are difficult to be quickly adapted to the target task trained on a small novel dataset. The challenges of the domain gap in image denoising are dataset domain gap and feature domain gap.

1) *Dataset Domain Gap:* The base dataset requires a large amount of data. Since the real-world noisy images contain a wide distribution and complex source of noise, it is difficult and time-consuming to collect such considerable data as a base dataset in real scenes. Following the common practices, synthetic noisy images are collected in our base dataset. However, traditional synthetic images produced from a single noise distribution cannot simulate the real noisy images well. Hence, there is a severe dataset domain gap between the novel and base datasets.

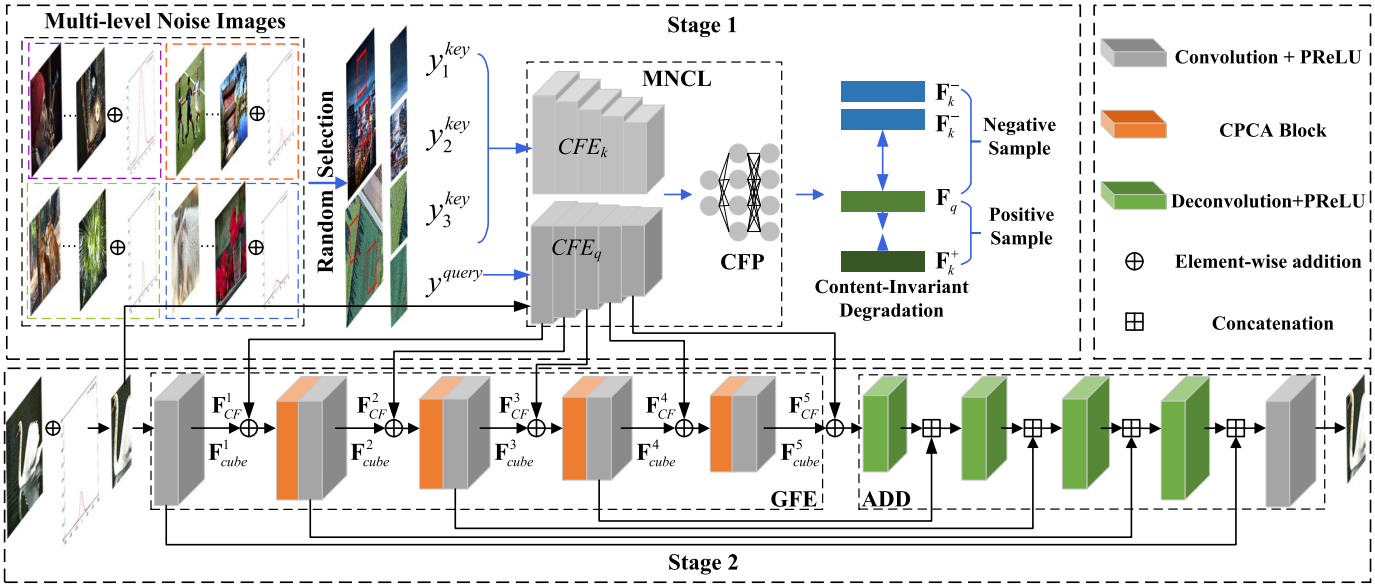


Fig. 1. Architecture of our proposed framework. The blue line represents the first training stage, and the black line represents the second training stage.

2) *Feature Domain Gap:* After training the network with the base dataset, how quickly adapting such synthetic noise features to the novel dataset is also a serious feature domain gap problem. An inappropriate interaction of features from widely different domains in a network may easily cause nonconvergence during training. Therefore, these two challenges of the domain gap in image denoising are urgent to solve.

Therefore, this work aims to address the problem of few-shot image denoising. Specifically, given a base dataset D_{base} composed of the synthetic noisy images with different additive white Gaussian noise (AWGN) levels and a novel dataset D_{novel} composed of a small number of noisy–clean image pairs, we then obtain a few-shot image denoising model through sequential training on D_{base} and D_{novel}. Such a few-shot image denoising model is meaningful for the widely practical denoising applications.

We propose a novel two-stage multilevel noise contrastive network (MNC-Net) for few-shot image denoising, as shown in Fig. 1. In the first stage, multilevel noise contrastive learner (MNCL) using contrastive learning and self-supervision learning is trained on the multiple-level noise D_{base} to ease the acute dataset domain gap. Specifically, the proposed MNCL is composed of the contrastive feature extractor (CFE) and contrastive feature projector (CFP). MNCL can learn rich and complex content-invariant degradations and general multiple-level noise representations by jointly using such two modules. Note that, in the process of the self-supervised training of the MNCL, a noisy image is used as the network input and its corresponding data augmented image is used as the supervision signal. At the same time, the CFE who undergo joint training in the first stage does not require retraining in the second stage. In the second stage, the CFE with frozen weights is first used to extract the corresponding contrastive features (CFs) from D_{novel}'s noisy images. At the same time, the guided feature encoder (GFE) uses our proposed combined pixel and channel attention (CPCA) block enhancing extract features at both pixel and channel levels from noisy images of D_{novel} to alleviate the feature domain gap between the two stages. Finally, the CFs and the features produced by GFE are fed into the adaptive denoising decoder (ADD) to adaptively denoise the images. Our proposed MNC-Net trained using only a few image pairs outperforms the baseline methods and achieves competitive performances compared to the state-of-the-art methods trained using much more times of image pairs. The contributions of our work are summarized as follows.

1) We first study the few-shot denoising strategy and propose a novel two-stage MNC-Net. Extensive experiments show that the proposed MNC-Net can avoid using massive noisy–clean image pairs to achieve better performance than the baseline methods and even compete with the state-of-the-art methods.
2) In the first stage, we collect multilevel noisy images in the base dataset called D_{base} to address the dataset domain gap. The MNCL, composed of the CFE and the CFP and trained on D_{base} using contrastive learning, is proposed to learn the rich and complex content-invariant degradations and then produces the general CFs for the subsequent learning phase.
3) In the second stage, we propose the GFE and the ADD to cope with the feature domain gap in the denoising network, which is trained on D_{novel}. The GFE guides the contrast features retrieved from CFE to produce the target domain representations for the specific input noisy images. Then, such representations are fed into the ADD to adaptively denoise the noisy images.

II. Related Work

A. General Image Denoising

In recent years, deep CNNs have become the mainstream of image denoising. Zhang *et al.* [3] proposed DnCNN

to learn the residual images. Supervising the training of thousands of noisy–clean paired images, DnCNN exceeds traditional image denoisers [13]–[17] for the denoising speed and peak signal-to-noise ratio (PSNR). Subsequently, numerous supervised denoising networks were proposed to improve the denoising performances [18]–[24]. However, these supervised image denoisers need large amounts of aligned noisy–clean image pairs for training. It is challenging and expensive to collect considerable noisy–clean images for supervised denoising. It hinders the effectiveness of applications with insufficient training pairs. Hence, some self-supervised denoisers were proposed to release this limit.

Lehtinen *et al.* [25] introduced Noise2Noise to train a deep denoiser with multiple independent noisy images from the same scenes. Inspired by Noise2Noise, several self-supervised denoising models were proposed to address the above problem. Examples include the Noise2Void [26] and Noise2Self [27] trained with noisy images in different scenarios. Notably, the carefully designed blind-spot networks are used for Noise2Self and Noise2Void to avoid learning the identity transformation. Noisier2Noise's [28] noise model generates the synthetic noise and adds such noise to the single noisy image for producing the training noisy image pairs, which is different from self-supervised methods. However, the noise model is hard to specify, especially in real-world scenarios. Noise-as-Clean [29] has a similar philosophy to Noiser2Noise. Quan *et al.* [30] proposed a Self2Self method trained to map between random dropout image pairs. The performance of this type of network deeply relies on the sampling mode.

B. Few-Shot Learning

Few-shot learning models trained using only a few training samples were proposed in [31] for object classification. Existing few-shot learning methods are dominated by meta-learning. They are divided into model-, metric-, and optimization-based methods. The model-based methods rapidly update parameters using a small number of samples and directly establish the mapping function from the input to the predicted values. Metric-based methods measure the distance between batch and support samples to classify the objects using the nearest neighbor. Because the general gradient descent methods are difficult to fit in the few-shot scenario, the optimization-based methods adjust the optimization to accomplish few-shot classification.

1) *Model-Based Methods:* Santoro *et al.* [9] proposed the method of memory enhancement to solve the few-shot learning task. The fast generalization ability of meta network [32] is derived from its "fast weigh" mechanism, and the gradient produced during training generates the "fast weigh."
2) *Metric-Based Methods:* Siamese network [10] trains the twin network in a supervised way and then reuses the features extracted from the network for one/few-shot learning. Compared with the twin network, the match network [33] constructs different encoders to train the support set and batch set. The final classifier produces the weighted summation of predicted values between the support set samples and queries.
3) *Optimization-Based Methods:* Ravi and Larochelle [11] uncovered update function or update scheme causes the failure of gradient-based optimization algorithms under a small number of training data. The method proposed by Finn *et al.* [34] makes it possible to obtain better generalization performance with a small number of iterations using a small number of samples. Also, the model is easy to fine-tune.

Previous few-shot works mainly tackle image classification tasks at a high level. Our work, however, focuses on image denoising at a low level. Hence, we cannot directly apply these previous meta-learning methods to the image denoising problem of few-shot learning. In our work, few-shot learning aims to recover potential clean images only using a small number of noisy–clean image pairs.

C. Contrastive Learning

Contrastive learning has proved its effectiveness in unsupervised representation learning. Previous methods [35]–[38] typically learn representations by minimizing the difference between the output and the fixed target. Contrastive learning does not use a predefined and fixed goal but maximizes mutual information in the representation space. Specifically, the representation of a query sample should attract positive counterparts while repelling negative counterparts. Transformed versions of the input [39]–[41], multiple views of the input [42], and neighboring patches in the same image [43], [44] can be considered positive counterparts.

In this article, image patches generated at the same Gaussian noise level are used as positive samples. The network obtains rich and complex content-invariant degradation representations and general representations on noisy images with multilevel noise by contrastive learning.

III. Proposed Method

Our framework, as shown in Fig. 1, consists of two training stages. This section first sets out the problem definition and introduces the overall framework. Next, we provide the training and the corresponding structures of the first and second stages in detail, respectively. Finally, the implementation details are illustrated.

A. Problem Definition and Overall Framework

1) Task: Few-shot image denoising aims to achieve satisfactory performance only utilizing limited noisy–clean image pairs.

2) Image Degradation Model: The noise model formulates noisy images as follows:

$$y = x + n \tag{1}$$

where x is a clean image, y is its corresponding noisy observation, and n refers to the noise.

3) Dataset Setting: Our method adopts the commonly used dataset name setup for few-shot learning, i.e., D_{base} and D_{novel} like [12].

Specifically, given a base dataset D_{base} $\{y_i = x_i + n_i\}_{i=1}^{800}$, where $\{n_i\}_{i=1}^{200} \sim \mathcal{N}(0, 5^2)$, $\{n_i\}_{i=201}^{400} \sim \mathcal{N}(0, 15^2)$, $\{n_i\}_{i=401}^{600} \sim \mathcal{N}(0, 25^2)$, $\{n_i\}_{i=601}^{800} \sim \mathcal{N}(0, 50^2)$, $\mathcal{N}(\mu, \sigma^2)$ is Gaussian distribution with expectation value μ and standard deviation σ.

Denote a novel dataset D_{novel} as $\{(y_i, x_i)\}_{i=1}^{N}$, where $N = 20$, 40, 60, or 80, y_i is a synthetic noisy image or a real-world noisy image, and x_i is the corresponding clean image. Meanwhile, $D_{\text{base}} \bigcap D_{\text{novel}} = \emptyset$.

Specific details of D_{base} and D_{novel} can be found in the datasets of Section IV-A.

4) Framework and Training Strategy: The training strategy of the proposed few-shot framework is divided into two stages. In the first training stage, the MNCL consisting of a CFE and a CFP learns rich and complex content-invariant degradations and general multilevel noise representations on multilevel noise D_{base} by contrastive learning. Then, the MNCL can perceive the different and complex noise distributions of the noisy images, which lessens the dataset domain gap problem. In the second training stage, the denoising network trained on D_{novel} is composed of a GFE and an ADD. It is worth noting that the CPCA block is further proposed in the GFE block. The weights of CFE are frozen. Next, the GFE induces the contrast characteristics produced from CFE to generate the target domain features of D_{novel}. Specifically, the proposed CPCA block can further enhance such features at pixel and channel levels to effectively mitigate the feature domain gap between the two stages. Potentially clean images can be finally recovered using ADD with the supervision of ground truth of D_{novel}.

B. Multilevel Noise Contrastive Learner

1) MNCL Structure: Given an image patch from D_{base} as the query patch ($y^{\text{query}} \in \mathbb{R}^{3\times256\times256}$ in Fig. 1, $y^{\text{query}} = x^{\text{query}} + n^{\text{query}}, n^{\text{query}} \sim \mathcal{N}(\mu^{\text{query}}, \sigma^{\text{query}2})$), other patches extracted from the same noise level of Gaussian distribution (e.g., y_3^{key}, where $y_3^{\text{key}} = x_3^{\text{key}} + n_3^{\text{key}}, n_3^{\text{key}} \sim \mathcal{N}(\mu_3^{\text{key}}, \sigma_3^{\text{key}^2})$, $\mu_3^{\text{key}} = \mu^{\text{query}}$, and $\sigma_3^{\text{key}} = \sigma^{\text{query}}$) can be considered as positive samples. In contrast, patches from other noise levels (e.g., y_1^{key} and y_2^{key}, where $y_1^{\text{key}} = x_1^{\text{key}} + n_1^{\text{key}}$ and $y_2^{\text{key}} = x_2^{\text{key}} + n_2^{\text{key}}, n_1^{\text{key}} \sim \mathcal{N}(\mu_1^{\text{key}}, \sigma_1^{\text{key}^2}), n_2^{\text{key}} \sim \mathcal{N}(\mu_2^{\text{key}}, \sigma_2^{\text{key}^2})$, and $\mu_1^{\text{key}} = \mu_2^{\text{key}} = \mu^{\text{query}}, \sigma_1^{\text{key}^2} = \sigma_2^{\text{key}^2} \neq \sigma^{\text{query}2}$) can be referred to as negative samples. Then, we encode the query using CFE_q with five convolution layers referring to the structure of stacked volume machine layers in image super-resolution [45] ($w_1 \in \mathbb{R}^{64\times3\times3\times3}$ and $w_i \in \mathbb{R}^{64\times64\times6\times6}$, where $i = 2, 3, 4, 5$) to obtain CFs, likewise encoder keys $y_i^{\text{key}} (i = 1, 2, 3)$ by CFE_k. Following the operation of SimCLR [40] and MoCo V2 [46], using an MLP projection head after the encoder can enhance the performance of model. CFs are fed to CFP consisting of two linear layers to obtain content-invariant degradations (i.e., $\{\mathbf{F}_q, \mathbf{F}_k^+, \mathbf{F}_k^-\} \in \mathbb{R}^{256}$ in Fig. 1). According to the similarity between content-invariant degradations, $\mathbf{F}_q$ is encouraged to close $\mathbf{F}_k^+$ while being away from $\mathbf{F}_k^-$. To measure the similarity of samples in the noise representation space, we use the contrast loss function, i.e., InfoNCE [43], adopted by MoCo v1 [41]. InfoNCE is reformulated as follows:

$$\mathcal{L}_{\mathbf{F}_q} = -\log \frac{\exp\left(\mathbf{F}_q \cdot \mathbf{F}_k^+ / \tau\right)}{\sum_{i=1}^{N} \exp\left(\mathbf{F}_q \cdot \mathbf{F}_k^- / \tau\right)} \tag{2}$$

where τ is a temperature hyperparameter and N is the number of negative samples.

The key to contrastive learning is to build a large dynamic dictionary covering considerable negative samples on the inputs. As suggested in MoCo v1 [41], the parameter θ_k of CFE_k is updated with momentum by

$$\theta_k = m\theta_k + (1 - m)\theta_q \tag{3}$$

where θ_q is the parameters of CFE_q and $m \in [0, 1)$ is a momentum coefficient. Maintain a queue containing samples with multilevel noise representations to obtain content-invariant degradations. Two patches are randomly cropped from each image of D_{base}, which contains N_{base} noisy images. The content-invariant degradations of $2N_{\text{base}}$ noisy image patches can be computed using the following equation:

$$\text{MNCL}\left(y_i^j\right) = \mathbf{F}_i^j \tag{4}$$

where y_i^j represents the jth patch of the ith image ($j \in \{1, 2\}$), $\mathbf{F}_i^j \in \mathbb{R}^{256}$ is its corresponding content-invariant degradation, and $\text{MNCL}(\cdot)$ is the complete network in the first training stage. For the ith noisy image, we refer to $\mathbf{F}_i^1$ and $\mathbf{F}_i^2$ as query and positive samples. The overall loss of the first training stage is defined as follows:

$$\mathcal{L}_{\text{degrad}} = \sum_{i=1}^{N_{\text{base}}} -\log \frac{\exp\left(\mathbf{F}_i^1 \cdot \mathbf{F}_i^2 / \tau\right)}{\sum_{j=1}^{N_{\text{queue}}} \exp\left(\mathbf{F}_i^1 \cdot \mathbf{F}_{\text{queue}}^j / \tau\right)} \tag{5}$$

where N_{base} is the number of noisy images in D_{base}, N_{queue} is the number of samples in the queue, and $\mathbf{F}_{\text{queue}}^j$ represents the jth negative sample.

2) Discussion: The MNCL learns rich and complex content-invariant degenerations by contrastive learning on multilevel Gaussian noisy images, which addresses the dataset domain gap problem. These content-invariant degenerations can also provide degeneration information for the feature extraction of real-world noisy images. Furthermore, we do not need the supervision of clean images. It proves that multilevel Gaussian noise is significantly effective for real-world image denoising in Section IV-D1.

C. Image Denoising Network

In the second training stage, an image denoising network, including GFE and ADD, is proposed to be trained on D_{novel}. Since almost no studies about few-shot image denoising, our network refers to the network structure of few-shot image classification, few-shot object detection, and so on. The core network structures of [7], [12], and so on usually use the popular backbones in their corresponding study area. Thus, we also use the UNet framework, a famous and popular structure in the denoising area, to construct our denoising network (GFE and ADD). To ease the feature domain gap of contrast characteristics and features of D_{novel}, the proposed

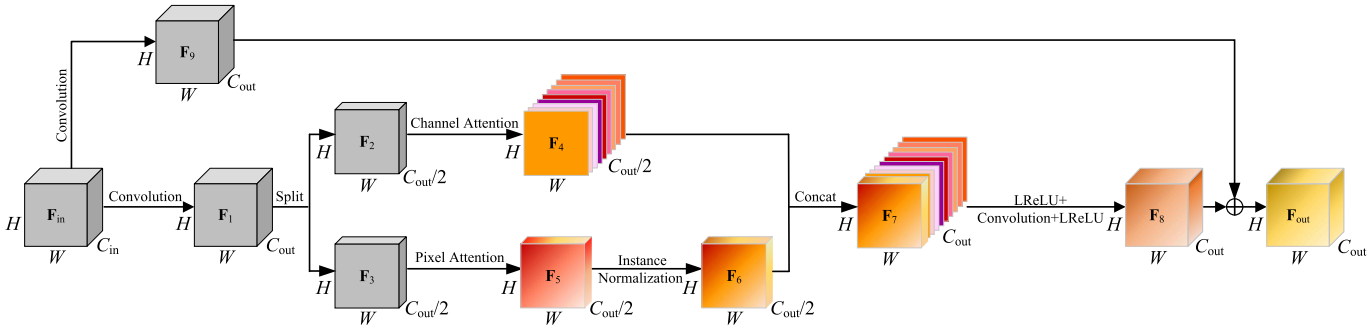


Fig. 2. Architecture of our proposed CPCA block.

CPCA block is added to the GFE to boost the feature extraction at pixel and channel levels on limited training data. Meanwhile, state-of-the-art methods are too complex and have more parameters to learn, which leads to high training time costs and consuming a lot of power. Thus, our network does not use other state-of-the-art method blocks. The following are specific descriptions of the second stage training network and training strategy.

1) Denoising Structure: Briefly, as shown in Fig. 1, the image denoising network is sequentially composed of GFE and ADD. Given an input noisy image $y \in \mathbb{R}^{3\times256\times256}$ that from D_{novel}, the trained CFE_q maps y to five CFs ($F^i_{CF} \in \mathbb{R}^{64\times(256/2^{i-1})\times(256/2^{i-1})}$, $i \in \{1,2,3,4,5\}$) using five convolution layers. At the same time, the GFE first maps y to a feature cube $\mathbf{F}^1_{\text{cube}} \in \mathbb{R}^{64\times256\times256}$ using a 3×3 convolution layer with the stride of 1 and the padding of 1. Then, F^1_{cube} is sequentially processed by the following four GFE's blocks to produce $\mathbf{F}^i_{\text{cube}} \in \mathbb{R}^{64\times(256/2^{i-1})\times(256/2^{i-1})}$ ($i \in \{2,3,4,5\}$). Specifically, every GFE's block sequentially stacks a CPCA block and a 6×6 convolution layer with the stride of 2 and the padding of 2. The number of channels is fixed to 64 across all blocks in GFE. All the convolution layers in GFE are followed by the elementwise addition operation $\mathbf{F}^i_{CF} + \mathbf{F}^i_{\text{cube}}$ ($i \in \{1,2,3,4,5\}$). Hence, the output of GFE is a feature cube $\mathbf{F}_{\text{cube}} \in \mathbb{R}^{64\times16\times16}$. The ADD contains four 6×6 deconvolution layers (stride is 2 and padding is 2) and a 3×3 convolution layer (stride is 1 and padding is 1). Each of the four deconvolution layers is followed by a concatenation operation. The last three deconvolution layers have 128 input channels and 64 output channels. The input and output channels in the first deconvolution layer are 64. Also, the convolution layer has 128 input channels and three output channels.

We denote $\hat{x} \in \mathbb{R}^{3\times256\times256}$ as the denoised output image of y and x as its corresponding ground-truth clean image. The loss function is reformulated as

$$\mathcal{L}_\Theta = \frac{1}{N_{\text{novel}}}\sum_{i=1}^{N}||\hat{x}_i - x_i|| \tag{6}$$

where Θ is the parameters of the denoising networks and N_{novel} is the pair's number of D_{novel}.

2) CPCA Structure: The CPCA proposes to retrieve the adaptive attention from both the channel and pixel extents for noisy images with a specific noise level.

Within each CPCA block, as shown in Fig. 2, the input feature $\mathbf{F}_{\text{in}} \in \mathbb{R}^{C_{\text{in}}\times H\times W}$ is first mapped to $\mathbf{F}_1 \in \mathbb{R}^{C_{\text{out}}\times H\times W}$ by a 3×3 convolution layer, where C_{in} and C_{out} are the numbers of input and output channels, respectively. $\mathbf{F}_1$ is then split into two parts ($\mathbf{F}_2, \mathbf{F}_3 \in \mathbb{R}^{C_{\text{out}}/2\times H\times W}$) in the channel dimension. Then, the first part $\mathbf{F}_2$ is transformed to $\mathbf{F}_4 \in \mathbb{R}^{C_{\text{out}}/2\times H\times W}$ using a squeeze-and-excitation (SE) block [47].

We use $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_{C_{\text{out}}/2}]$ to denote the SE block learned set of filter kernels, where $\mathbf{v}_c$ refers to the parameters of the cth filter. We can write $\mathbf{F}_4 = [\mathbf{f}^1_4, \mathbf{f}^2_4, \ldots, \mathbf{f}^{C_{\text{out}}/2}_4]$, where

$$\mathbf{f}^c_4 = \mathbf{v}_c * \mathbf{F}_2 = \sum_{i=1}^{C_{\text{out}}/2}\mathbf{v}^i_c * \mathbf{f}^i_2. \tag{7}$$

Here, $*$ denotes convolution, $\mathbf{v}_c = [\mathbf{v}^1_c, \mathbf{v}^2_c, \ldots, \mathbf{v}^{C_{\text{out}}/2}_c]$, $\mathbf{F}_2 = [\mathbf{f}^1_2, \mathbf{f}^2_2, \ldots, \mathbf{f}^{C_{\text{out}}}_2/2]$, and $\mathbf{f}^c_2 \in \mathbb{R}^{H\times W}$. $\mathbf{v}^i_c$ is a 2-D spatial kernel that represents a single channel of $\mathbf{v}_c$. $\mathbf{v}^i_c$ acts on the corresponding channel of $\mathbf{F}_2$. The second part $\mathbf{F}_3$ gets $\mathbf{F}_6 \in \mathbb{R}^{C_{\text{out}}/2\times H\times W}$ that uses the pixel attention (PA) [48] operation following the instance normalization (IN) [49]:

$$\mathbf{F}_6 = \text{IN}(\text{sigmod}(\omega_{1\times1} * \mathbf{F}_3)\cdot\mathbf{F}_3) \tag{8}$$

where $\mathbf{F}_5 = \text{sigmod}(\omega_{1\times1} * \mathbf{F}_3)\cdot\mathbf{F}_3$, $\omega_{1\times1} \in \mathbb{R}^{(C_{\text{out}}/2)\times(C_{\text{out}}/2)\times1\times1}$ and $\text{IN}(\cdot)$ computes the mean and standard deviation (std) along the (H, W) axes to each sample at the spatial extent. Next, $\mathbf{F}_4$ and $\mathbf{F}_6$ are concatenated along the channel dimension to produce $\mathbf{F}_7 \in \mathbb{R}^{C_{\text{out}}\times H\times W}$. Then, the residual features $\mathbf{F}_8 \in \mathbb{R}^{C_{\text{out}}\times H\times W}$ are obtained by feeding $\mathbf{F}_7$ to one 3×3 convolution and two LReLU layers

$$\mathbf{F}_8 = \text{LReLU}(\omega_{3\times3} * \text{LReLU}(\mathbf{F}_7)) \tag{9}$$

where $\omega_{3\times3} \in \mathbb{R}^{C_{\text{out}}\times C_{\text{out}}\times3\times3}$ and $\text{LReLU}(\cdot)$ represents the LReLU layer. The shortcut feature $\mathbf{F}_9 \in \mathbb{R}^{C_{\text{out}}\times H\times W}$ is generated by $\mathbf{F}_{\text{in}}$ through the 1×1 convolution

$$\mathbf{F}_9 = \omega_{1\times1} * \mathbf{F}_{\text{in}} \tag{10}$$

where $\omega_{1\times1} \in \mathbb{R}^{C_{\text{out}}\times C_{\text{out}}\times1\times1}$. Finally, the CPCA block produces $\mathbf{F}_{\text{out}} \in \mathbb{R}^{C_{\text{out}}\times H\times W}$ by adding $\mathbf{F}_8$ and the shortcut feature $\mathbf{F}_9$. $\mathbf{F}_{\text{out}}$ can be reformulated as

$$\mathbf{F}_{\text{out}} = \mathbf{F}_8 + \mathbf{F}_9. \tag{11}$$

3) Discussion: By utilizing the rich and complex content-invariant degradations learned in the first stage, our denoising network requires only a few paired noisy–clean images to quickly adapt to the specific noise level to generate the corresponding distribution denoised images. Hence, we fuse the CFs extracted by CFE and features extracted

TABLE I

AVERAGE PSNR AND SSIM RESULTS OF BASELINES AND OUR NETWORK ON CBSD68, KODAK24, AND SET12 WITH NOISE LEVELS OF 15, 25, AND 50

Methods	Shot	$\sigma = 15$						$\sigma = 25$						$\sigma = 50$					
		CBSD68		Kodak24		Set12		CBSD68		Kodak24		Set12		CBSD68		Kodak24		Set12	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
RIDNet-ft	20	31.14	0.8579	30.96	0.8267	27.58	0.8035	27.91	0.7520	27.90	0.7051	26.47	0.7097	25.65	0.6474	25.99	0.6151	24.87	0.6243
	40	28.65	0.7314	28.54	0.6846	27.61	0.7109	28.83	0.7938	29.14	0.7631	27.39	0.7520	25.72	0.6549	26.10	0.6241	25.01	0.6363
	60	26.42	0.6345	26.34	0.5802	26.20	0.6330	28.27	0.7623	28.46	0.7236	27.00	0.7286	25.77	0.6600	26.16	0.6291	25.15	0.6433
	80	24.79	0.5650	24.74	0.5076	24.89	0.5762	25.64	0.6137	25.67	0.5573	25.23	0.6009	24.49	0.5687	24.75	0.5276	24.12	0.5650
	400	24.80	0.5652	24.75	0.5076	24.89	0.5760	24.83	0.5728	24.72	0.5123	24.41	0.5549	25.92	0.6599	26.49	0.6408	25.38	0.6357
MPRNet-ft	20	30.89	0.8459	31.01	0.8074	29.44	0.8322	28.01	0.7608	28.26	0.7164	27.10	0.7498	24.76	0.6284	25.03	0.5817	24.06	0.6262
	40	31.08	0.8590	31.29	0.8232	29.71	0.8448	28.31	0.7843	28.57	0.7409	27.32	0.7668	24.98	0.6558	25.37	0.6192	24.19	0.6522
	60	31.31	0.8684	31.54	0.8342	29.89	0.8514	28.45	0.7953	28.77	0.7562	27.46	0.7769	25.00	0.6538	25.36	0.6143	24.28	0.6497
	80	31.42	0.8725	31.64	0.8388	29.95	0.8549	28.63	0.8010	28.93	0.7616	27.62	0.7824	25.10	0.6727	25.50	0.6350	24.37	0.6626
	400	33.25	0.9189	33.54	0.9015	31.19	0.8676	30.60	0.8704	31.10	0.8558	29.34	0.8190	27.16	0.7783	27.80	0.7630	26.48	0.7509
Ours	20	33.09	0.9147	33.17	0.8941	30.86	0.8509	30.33	0.8618	30.75	0.8474	28.82	0.8053	26.88	0.7578	27.56	0.7541	25.77	0.7124
	40	33.35	0.9189	33.71	0.9071	31.43	0.8650	30.56	0.8658	31.13	0.8549	29.34	0.8198	27.29	0.7669	27.94	0.7595	26.47	0.7313
	60	33.62	0.9252	33.87	0.9097	31.63	0.8699	30.82	0.8719	31.35	0.8598	29.62	0.8257	27.47	0.7698	28.15	0.7669	26.80	0.7420
	80	**33.70**	**0.9254**	**33.92**	**0.9104**	**31.77**	**0.8723**	**31.01**	**0.8773**	**31.53**	**0.8644**	**29.78**	**0.8307**	**27.68**	**0.7789**	**28.35**	**0.7719**	**26.96**	**0.7517**

by GFE. The addition of CFs from the first stage makes our network not need to rely on a large amount of noisy–clean image pairs for training in the second stage. When real-world noisy images are not included in D_{base}, a satisfactory denoising effect can be produced by only training on a small number of real-world noisy–clean image pairs through the second training stage.

D. Implementation Details

1) Training Details: Throughout the network, all PReLU activation function hyperparameters are 0.2. The Adam optimizer is utilized for training. The learning rate is decayed exponentially from $1e^{-4}$ to $1e^{-6}$ for 8k iterations. In the contrastive learning, $m = 0.999$ in (3), and the temperature hyperparameter τ in (5) is set to 0.0100 and $B = 32$. All experiments are conducted on a server with Python 3.7.10, PyTorch 1.8, and an Nvidia-SMI GeForce RTX 3090 GPU.

2) Inference Details: Feed the noisy image into CFE_q and GFE. Then, the ADD produces the restored image.

IV. EXPERIMENT

A. Experimental Setup

1) Datasets: Three widely used image denoising datasets are selected to train our framework: DIV2K [50], BSD500 [51], and SIDD [52].

D_{base}: DIV2K includes 800 color images. They are corrupted by AWGN with noise levels: $\sigma = 5/15/25/50$. We randomly select four groups of images from DIV2K. Each group has 200 images, and the noise level is $\sigma = 5/15/25/50$ in sequence. The four groups of images do not overlap and are collectively termed as D_{base}.

D_{novel}: BSD500 [51] consists of 400 color images. They are corrupted by AWGN with noise levels: $\sigma = 15/25/50$. We randomly select $20/40/60/80$ pairs of noisy–clean images for every noise level in BSD500. SIDD [52] consists of 320 pairs of real noisy images and corresponding clean images for training. We randomly select $20/40/60/80$ pairs. These noisy–clean image pairs from BSD500 and SIDD are called $D_{\text{novel}}^{\text{BSD}}$ and $D_{\text{novel}}^{\text{SIDD}}$, respectively.

We evaluate our proposed framework and competitive methods on synthetic and real datasets. Three datasets are used for the performance evaluation in the case of AWGN, including CBSD68 [53] with 68 color images, Kodak24 [54] with 24 color images, and Set12 [3] with 12 color images. Images are corrupted by the AWGN with noise levels: $\sigma = 15, 25, 50$ for Set12, CBSD68, and Kodak24, respectively. The 1280 image pairs for validation in SIDD [52] and 50 pairs of real-world noisy and noise-free scenes in DND [55] are used for the performance evaluation of real noise.

2) Baselines: We compare our framework with two competitive baselines. Both of them are supervised methods for real-world image denoising. The first baseline is built on RIDNet [5] and is called RIDNet-ft. The second one is built upon MPRNet [56] and is termed as MPRNet-ft. They are trained on D_{novel} and the full 400 image pairs of BDS500. Since clean images are not used in our network and these two methods are based on supervision, the training datasets of these two baselines do not include D_{base}. For a fair comparison, we use the default settings of RIDNet and MPRNet provided by the corresponding literature. Comparing the two baselines can further understand the few-shot learning advantages of our method.

B. Comparisons With Baselines

To validate the effectiveness of our method on synthetic and real-world noisy images, the proposed method first compares to baselines under the same few-shot training strategy, i.e., using the same small number of noisy–clean images to train all the compared methods.

1) Synthetic Noisy Images: Table I shows the PSNR and SSIM values between our method and two baselines on the synthetic noisy image datasets: CBSD68, Kodak24, and Set12. As can be seen from the table, the method outperforms the baselines at all corresponding shot levels and noise levels for all three datasets. When the number of training pairs is only $20/40/60/80$, the performances of RIDNet-ft and MPRNet-ft are significantly lower than those of our method. In particular, compared to MPRNet-ft, the proposed method improves the performances by 2.20, 2.32, and 2.12 dB on the CBSD68 dataset with a shot level of 20 on noise levels of 15, 25,

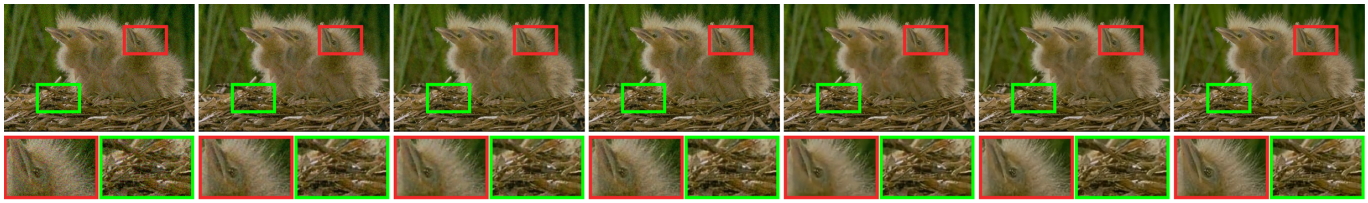


Fig. 3. Visual comparisons between our network and its baselines on the CBSD68 dataset with a noise level of $\sigma = 15$.

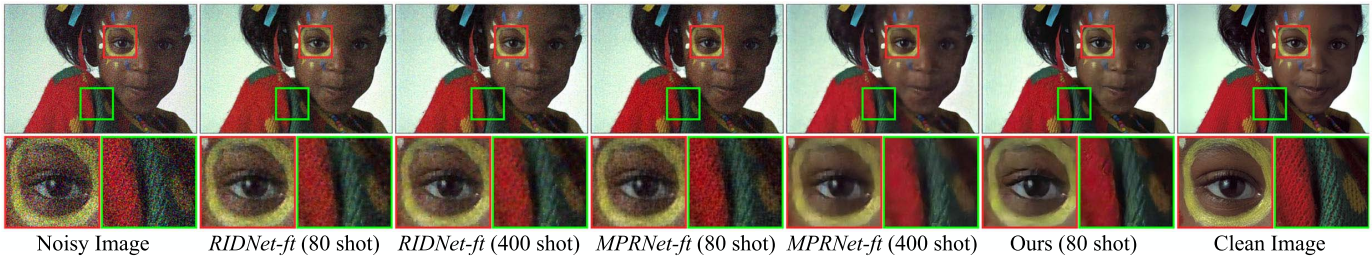


Fig. 4. Visual comparisons between our network and its baselines on the Kodak24 dataset with a noise level of $\sigma = 50$.

TABLE II
AVERAGE PSNR AND SSIM RESULTS OF BASELINES AND OUR NETWORK ON THE SIDD DATASET

Shot	Methods					
	RIDNet-ft		*MPRNet-ft*		Ours	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
20	31.67	0.6913	33.69	0.8757	**38.15**	**0.9012**
40	33.42	0.7591	34.45	0.8865	**38.40**	**0.9056**
60	32.07	0.7107	34.98	0.8946	**38.57**	**0.9067**
80	28.85	0.5705	35.09	0.8928	**38.61**	**0.9070**

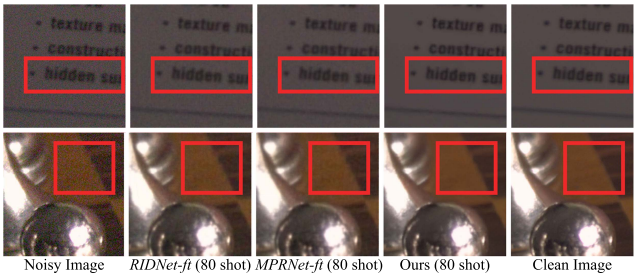


Fig. 5. Visual comparisons between our network and its baselines on the SIDD dateset.

and 50, respectively. Such results are also consistent with those of other corresponding comparison settings. Furthermore, the PSNR produced by our method trained using 40 pairs of noisy–clean images is almost higher than those of training RIDNet-ft and MPRNetft using all the images of datasets (400 pairs of noisy–clean images), e.g., 8.55 and 0.10 dB higher, respectively, on the CBSD68 with noise level $\sigma = 15$.

Furthermore, the visual comparisons of the baseline methods and our method on CBSD68 and Koadk24 datasets are shown in Figs. 3 and 4, respectively. From Figs. 3 and 4, a closer inspection of the chicken's eye and the children's eye reveals that our method generates the closest textures to the clean image with more details compared to the baseline methods.

All the above comparisons demonstrate the effectiveness of the proposed method on the synthetic noisy image datasets. When training our method and general denoising methods with limited pairs of noisy–clean images, the few-shot superiority of our method can be clearly demonstrated. The metric of our method is that it can achieve satisfactory results with uncomplicated CNNs on a limited training set. This advantage is not available in the previous methods.

2) Real-World Noisy Images: To further assess the generalization of our model, we compare the proposed method to the baselines on real noisy images. Table II presents the quantitative results (PSNR) and SSIM of baselines and our network on the real noisy dataset SIDD. Our method produces much better performances than the baseline methods on all the comparison settings. For example, the proposed method outperforms RIDNet-ft and MPRNet-ft by a large margin of 9.76 and 3.52 dB when using 80 pairs of D_{novel}, respectively.

Fig. 5 compares the denoised results between a text image and a scene image on the SIDD dataset. Our method recovers the colors, which are much closer to the original pixel values, while baselines cannot satisfactorily restore original colors. All the above results demonstrate the effective generalization of our network on real noisy images with only a few labeled image pairs. Our method not only achieves good denoising performance on synthetic noisy datasets but also exhibits strong denoising ability on real noisy datasets. Our method also shows good generalization when encountering noise not present in D_{base}. Therefore, this proves the effectiveness of our method in solving the domain gap problem.

The proposed method performs the best in the above comparisons of baselines on synthetic and real-world noisy images in Tables I and II. This indicates the outstanding denoising capability of the proposed method.

C. Comparisons to Competitive State-of-the-Art Methods

To further comprehensively evaluate our method, the proposed method trained under the few-shot training strategy

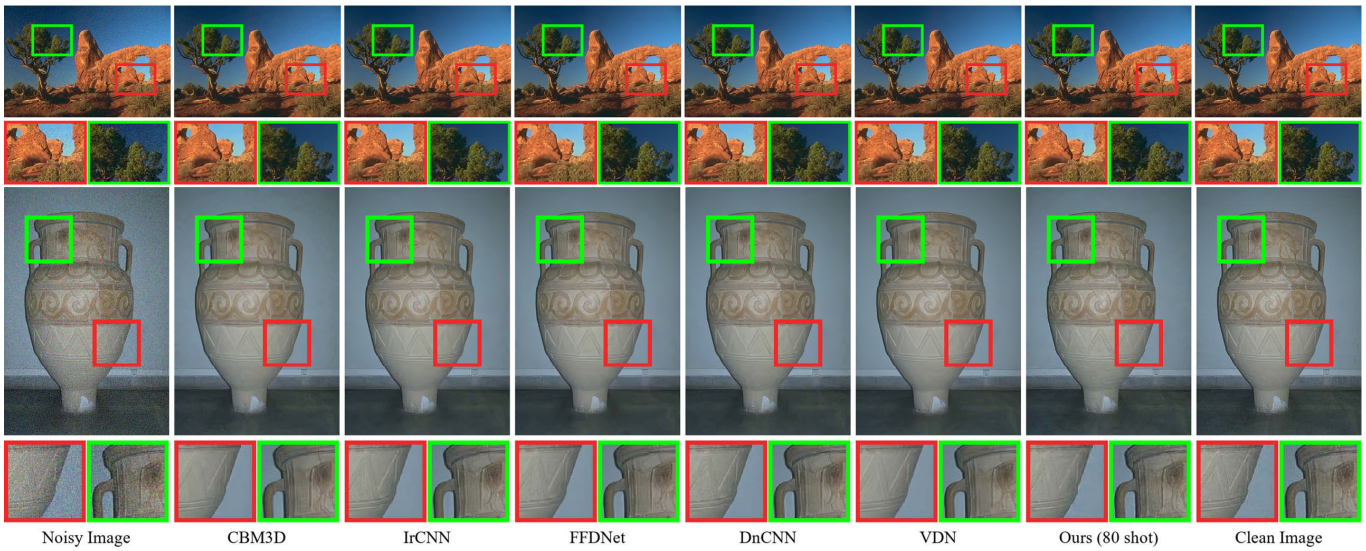


Fig. 6. Visual comparisons between our network and competitive state-of-the-art methods on the CBSD68 dataset with a noise level of $\sigma = 15$.

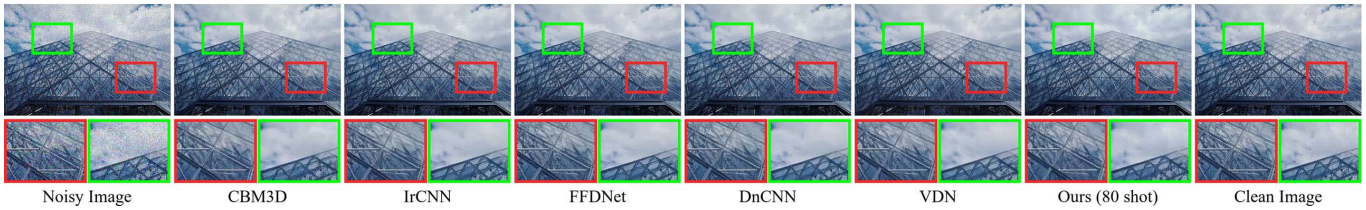


Fig. 7. Visual comparisons between our network and competitive state-of-the-art methods on the CBSD68 dataset with a noise level of $\sigma = 25$.

TABLE III

AVERAGE PSNR AND SSIM RESULTS OF DIFFERENT METHODS AND OUR NETWORK ON THE CBSD68 WITH NOISE LEVELS OF 15, 25, AND 50

Method	CBM3D		MLP		TNRD		CNLNet		IrCNN		FFDNet		DnCNN		VDN		RIDNet		**Ours**	
Image pairs	None		150000		400		400		5544		5544		400		5000		4000		**80**	
Noise levels	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
$\sigma = 15$	33.50	0.9211	-	-	31.37	-	33.69	-	33.86	0.9338	33.87	0.9330	33.89	0.9338	33.90	**0.9358**	**34.01**	-	33.70	0.9254
$\sigma = 25$	30.69	0.8752	28.92	-	28.88	-	30.96	-	31.16	0.8894	31.21	0.8879	31.33	0.8891	31.35	**0.8906**	**31.37**	-	31.01	0.8773
$\sigma = 50$	27.37	0.7723	26.00	-	25.94	-	27.64	-	27.86	0.7992	27.96	0.7965	27.97	**0.7979**	**28.19**	0.7883	28.14	-	27.68	0.7789

is compared to the state-of-the-art methods trained under the regular training strategies from the corresponding references, i.e., CBM3D [57] and CNN-based supervised denoisers, including MLP [58], TNRD [59], DnCNN [3], IrCNN [60], CNLNet [61], FFDNet [20] CBDNet [23], RIDNet [5], and VDN [4].

1) Synthetic Noisy Images: In this section, we compare against state-of-the-art competitive methods on synthetic images corrupted by AWGN. Table III reports the PSNR and SSIM results between our network and the compared supervised methods on the CBSD68 dataset. It is worth noting that the satisfactory compared methods use about five or even thousands of times image pairs for training compared to our method, while our method only uses 80 pairs of noisy–clean image pairs for training. In Table III, our network performs much better than the traditional CBM3D method to a large extent. For example, when $\sigma = 25$, the PSNR produced by our method is 0.32 dB higher than that of CBM3D. Our network also surpasses three supervised methods, i.e., 1.68, 1.74, and 0.04 dB higher than MLP, TNRD, and CNLNet on the CBSD68 dataset with a noise level of 50, respectively. Our method is slightly inferior to the remaining methods but still achieves the competitive results compared to the best VDN and RIDNet. The gap between our method and the best RIDNet is tiny at 0.36 dB with a noise level of 25.

Such PSNR and SSIM reductions are paltry for the visualization comparisons, proved in Figs. 6 and 7, Figs. 6 and 7 show the visual comparisons of different denoising methods. Our recovered image is almost indistinguishable from the other four methods in visual effect. By viewing the enlarged image, the recovered image produced from VDN still produces some oversmoothing effect in the details. However, the recovered image from our network retains more detailed information. These fully prove the advantages and effectiveness of our few-shot network. Based on baseline comparison, the experiments in this section further prove that our few-shot-based image denoising method can still compete with state-of-the-art methods.

2) Real-World Noisy Images: In addition, for real-world noisy image denoising, we also evaluate our network and

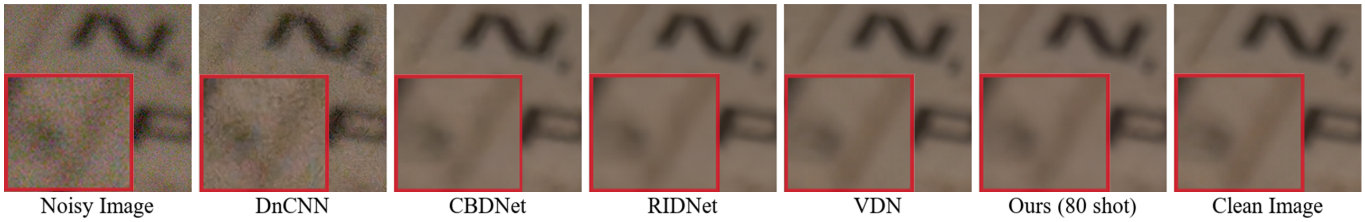


Fig. 8. Visual comparisons between our network and competitive state-of-the-art methods on the SIDD dataset.

TABLE IV
AVERAGE PSNR AND SSIM RESULTS OF DIFFERENT METHODS AND OUR NETWORK ON THE SIDD AND DND DATASETS

Dataset		SIDD		DND	
Method	Image pairs	PSNR	SSIM	PSNR	SSIM
DnCNN	400	23.66	0.583	32.43	0.790
MLP	150000	24.71	0.641	34.23	0.833
BM3D	None	25.65	0.685	34.51	0.851
FFDNet	5544	29.20	-	34.40	0.847
CBDNet	3720	30.78	0.801	38.06	0.942
RIDNet	480	38.71	0.951	39.26	0.953
VDN	320	**39.28**	**0.956**	**39.38**	**0.952**
Ours	**80**	38.61	0.907	38.67	0.944

TABLE V
AVERAGE PSNR AND SSIM RESULTS IN TERMS OF TRAINING MNCL WITH DIFFERENT NOISE-LEVEL COMBINATIONS OF D_{BASE} ON SIDD

	SIDD	
Noise levels	PSNR	SSIM
None	36.02	0.8988
$\sigma = 50$	38.48	0.9064
$\sigma = 25, 50$	38.56	0.9066
$\sigma = 15, 25, 50$	38.59	0.9067
$\sigma = 5, 15, 25, 50$	**38.61**	**0.9070**

other state-of-the-art competitive methods. Table IV presents the quantitative results (PSNR and SSIM) of several image denoising methods on SIDD and DND datasets. It is worth noting that these methods use about four or even thousands times image pairs for training compared to ours only 80 training noisy–clean image pairs. From Table IV, the proposed network is superior to the first five methods and competes with RIDNet and VDN on SIDD and DND datasets.

Fig. 8 compares the visualization of the denoised images produced by the state-of-the-art and our proposed method. It is apparent that our network can effectively remove the real noise and restore the true color of the image at the same time, producing satisfactory denoised images to the ground truth. However, DnCNN removes the noise unsatisfactorily, and there is still a little residual noise in the CBDNet recovered image. Notably, our method achieves an almost indistinguishable visual effect compared to RIDNet and VDN. In particular, the naked eye almost cannot tell the difference even in the enlarged image. In Table IV, our method is slightly inferior to RIDNet in PSNR and SSIM, e.g., 0.10 dB lower on the SIDD dataset. Our method, however, reduces the training data by six times. Such indistinguishable difference exactly proves the effectiveness of our method. It is probably due to the variety of content-invariant degradation learned in the first stage using contrast learning on multilevel noise and the enhanced features retrieved by the proposed CPCA block at pixel and channel level in the second stage. Our method effectively solves the domain gap problem. This produces a better generation for multilevel noise and results in more satisfactory denoised images when processing the real noisy images with complex noise sources.

According to the comprehensive evaluations on synthetic and real-world noisy images, our network can achieve a better tradeoff between the quantitative and visualization results. Though VDN and RIDNet are slightly superior to our method, they, respectively, use 5000 and 4000 synthetic image pairs or 320 and 480 real-world noisy image pairs to train the networks, while only 80 noisy–clean image pairs are used to train our method. Therefore, our method significantly reduces about 50–60 times synthetic training image pairs or 4–6 times real-world training image pairs with an acceptable tolerance of performance reduction. This is consistent with our goal of the few-shot method for training using limited data while producing a satisfactory performance. This further demonstrates not only an effective ability of our few-shot denoising method but also a high potential in performance improvement for future few-shot denoising studies.

D. Ablation Studies

We verify the effects of various components in our framework by comparing the number of noise levels, the number of noisy–clean image pairs, the temperature hyperparameter of contrastive learning, the performance of the CPCA block, and the other image degradation models.

1) Number of Noise Levels: In the proposed method, D_{base} consisting of multilevel noisy images can softly address the dataset domain gap. MNCL learns the general multilevel noise representations on D_{base}. Because the number of noise levels contained in D_{base} directly affects the generalization of the learned multilevel noise representations produced by the MNCL, we conduct experiments to train MNCL using D_{base} with various combinations of noise levels. Table V presents the PSNR and SSIM results among training MNCL on different noise combinations of D_{base}. It can be clearly observed in Table V that, when there are no noise levels, i.e., without the MNCL and the elementwise addition operations in the GFE, our denoising performance decreases sharply. This verifies the effectiveness of the MNCL and elementwise operations, which

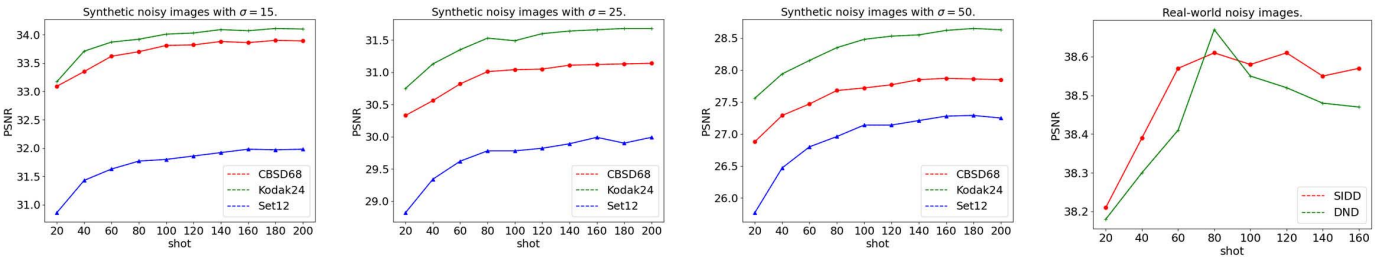


Fig. 9. Average PSNR and SSIM comparisons from our method trained under multiple numbers of noisy–clean image pairs on synthetic datasets and real-world datasets.

Fig. 10. Classification comparisons of content-invariant degradation feature produced from the first training stage under five different τ values.

is probably because such two operations can generate the general degradation representation to better ease the dataset domain gap. Hence, with the increasing number of noise levels, the performance produced by the proposed method is also improved. However, the training computation cost is always constant because the number of noisy images contained in D_{base} is always 800. It further proves that MNCL can learn the representations with a wide range of noise distributions and further ease the dataset domain gap due to the multilevel noise contained in the noisy images. Thus, such multilevel noise representation probably contributes to the satisfactory performances of real-world noisy image denoising. Experiments in Table II also prove this point. In Table V, since the performance draws to saturate when using four noise levels and more noise levels increase the difficulty of contrast learning, we use these four noise levels in our method to achieve an appropriate tradeoff between performance and difficulty.

2) Number of Noisy–Clean Image Pairs: According to [58], CNN-based supervised denoisers trained on a large amount of training data achieve better denoising results than training on a small amount of training data. However, our work dedicates to few-shot image denoising using a small number of supervised training image pairs. Hence, the number of image pairs directly affects the final denoising results of our method. We conduct experiments on multiple small numbers of noisy–clean image pairs on CBSD68, Kodak24, Set12, SIDD, and DND datasets to seek the proper amount of training data.

As shown in Fig. 9, the PSNR approaches saturation on CBSD68, Kodak24, and Set12 datasets when the number of noisy–clean image pairs reaches 80, especially when the noise levels are $\sigma = 15$ and 25, while the best performance is achieved using 80 image pairs on the real-world noisy dataset. To ensure the efficiency of few-shot image denoising and achieve competitive denoising results, we stop at 80 pairs of noisy–clean images. The experiments in Tables I and II also prove that using 80 training image pairs can achieve satisfactory results in extensive comparisons.

TABLE VI

AVERAGE PSNR AND SSIM RESULTS IN TERMS OF τ ON CBSD68, KODAK24, AND SET12

τ	CBSD68		Kodak24		Set12	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
0.0010	33.70	0.9265	33.92	0.9097	31.67	0.8723
0.0100	**33.70**	**0.9882**	**33.92**	**0.9115**	**31.77**	**0.8725**
0.1000	33.63	0.9282	33.85	0.9113	31.68	0.8714

3) Temperature Hyperparameter: τ influences the distribution of content-invariant degradations in the noise representation space in (2) and (5). To prevent all the representations gathered in one place, i.e., collapse, we compare different τ values. Fig. 10 shows the different classification results under different τ values. When τ is equal to 0.0001 or 1.0000, the images with four noise levels are still mixed. However, the other three values of τ produce significant classifications. In order to achieve the best denoising performance, Table VI compares the PSNR and SSIM values after completing training two stages under the three values of τ. It is clear that the optimal denoising effect occurs when $\tau = 0.0100$. This is most probably caused by the following reason.

InfoNCE is a loss function that can perceive the difficulty of negative samples using τ. In the training process of our method, noise levels and contents of noisy images for learning contrast features are both different. This undoubtedly increases the difficulty of perceiving negative samples. In this case, τ is usually set to a relatively smaller value to achieve satisfactory classifications. However, if τ is too small, some difficult negative samples will be clustered to the positive examples, destroying the final classifications of positive and negative examples. Therefore, in Table VI, $\tau = 0.0100$ may be an appropriate value for our model.

TABLE VII
AVERAGE PSNR AND SSIM RESULTS IN TERMS OF THE CPCA BLOCK ON CBSD68, KODAK24, AND SET12

Cases	CPCA-CFE	CPCA-GFE	CBSD68		Kodak24		Set12	
			PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
a	✗	✗	33.61	0.9243	33.80	0.9075	31.55	0.8671
b	✓	✗	33.54	0.9235	33.70	0.9063	31.47	0.8655
c	✗	✓	**33.70**	**0.9882**	**33.92**	**0.9115**	**31.77**	**0.8725**
d	✓	✓	33.66	0.9252	33.86	0.9091	31.78	0.8704

4) Performance of CPCA Block: In our framework, we apply the CPCA block to enhance the quality of extracted features from the GFE block. Then, the feature domain gap is alleviated after fusing the CFs extracted by CFE and features extracted by GFE. We conduct experiments on various versions of models with or without the CPCA block in CFE and GFE, as shown in Table VII. The comparisons are performed on CBSD68, Kodak24, and Set12 datasets using our network trained on 80 pairs of noisy–clean images ($\sigma = 15$) and the temperature hyperparameter $\tau = 0.0100$. In Table VII, first, from the comparison pairs of Cases a versus b and Cases c versus d, using the CPCA method in the CFE method will cause performance degradation. This is probably because CPCA may limit the CFE's expression, destroying the learning generalization on multilevel noise, which contradicts the mechanism of CFE. Second, according to the comparison pairs of Cases a versus c and Cases b and d, utilizing the CPCA method in the GFE can effectively improve the performance of the proposed model. The reason is that CPCA can generate adaptive attention to retrieve the features at the specific noise level, leading to a better narrowing of the feature domain gap and guidance for recovering from the noisy images. Finally, according to all the comparisons, the CPCA method is only employed in the GFE module in the proposed method.

5) Comparison on Other Image Degradation Models: To demonstrate the generality of our model, we conducted experiments using the impulse noise and speckle noise. The training strategy is the same as AWGN. Specifically, the first stage network is trained on D_{base} and the second stage network is trained on D_{novel}. In addition, our method is also compared with RIDNet and VDN, which performs the best on real-world noisy images. Since RIDNet and VDN do not perform denoising experiments on impulse and speckle noise, we retrain them on impulse and speckle noise using the training settings of synthetic noise from their literature.

The training data with impulse and speckle noise are constructed as follows. When constructing impulse noisy image data, each pixel of clean images is replaced by a uniformly sampled random color in $[0, 1]^3$ with a specific probability α [62]. In the experiments of denoising impulse noise, D_{base} consists of four levels of α in noisy images, which are 0.05, 0.10, 0.15, and 0.20. The 800 color images of DIV2K are corrupted by these four levels of α to establish D_{base}. The 400 color images from BSD500 are corrupted by the four levels of α. We randomly select 80 noisy images and their corresponding clean images from such 400 corrupted images to build D_{novel}. When constructing the speckle noisy image data, speckle noise is typically modeled as multiplicative noise, which is formulated as $y = x + n * x$, where n is the random noise sampled from a uniform distribution with mean 0 and variance s [63], x is a clean image, and y is its corresponding noisy observation. D_{base} consists of 800 color images of DIV2K corrupted by four levels of s, i.e., $s = 0.01/0.05/0.10/0.15$. The 400 color images from BSD500 are corrupted by the four levels of s. We randomly select 80 noisy images and their corresponding clean images from such 400 corrupted color images to form D_{novel}.

TABLE VIII
AVERAGE PSNR AND SSIM RESULTS OF DIFFERENT METHODS ON THE CBSD68 WITH IMPULSE NOISE'S α OF 0.05, 0.10, 0.15, AND 0.20 AND SPECKLE NOISE'S s OF 0.01, 0.05, 0.10, AND 0.15

Method		RIDNet		VDN		Ours	
Image pairs		4000		5000		80	
Noise type	Noise level	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Impulse Noise	$\alpha = 0.05$	38.58	0.9760	38.84	0.9803	**39.62**	**0.9805**
	$\alpha = 0.10$	37.03	0.9623	36.76	0.9610	**37.56**	**0.9706**
	$\alpha = 0.15$	32.32	0.9406	32.18	0.9344	**33.39**	**0.9485**
	$\alpha = 0.20$	30.12	0.8868	31.00	**0.9321**	**31.95**	0.9277
Speckle Noise	$s = 0.01$	35.55	0.9575	**36.39**	**0.9613**	35.61	0.9578
	$s = 0.05$	32.00	0.9400	**32.16**	**0.9406**	31.06	0.9009
	$s = 0.10$	**30.02**	**0.8819**	29.76	0.8698	29.34	0.8631
	$s = 0.15$	27.79	0.8327	26.95	0.8248	**28.44**	**0.8364**

Table VIII shows the quantization results of removing impulse and speckle noise within all the compared models. Our method performs the best in almost all the comparisons, which demonstrates that our proposed strategy is effective in removing such two kinds of noise using only 80 training image pairs. Specifically, for impulse noise, our method outperforms RIDNet and VDN by 1.07 and 1.21 dB in PSNR at $\alpha = 0.15$, respectively. On the datasets using speckle noise, our method achieves the best results at the largest noise intensity with $s = 0.15$. This can prove the effectiveness of our method on large noise intensity. On other noise levels of s, our method also achieves competitive results compared to other methods. It is mainly because the noise intensity affects RIDNet and VDN more than our method. When s is increased from 0.01 to 0.15, our results only drop by 7.17 dB, while RIDNet and VDN drop by 7.67 and 9.44 dB, respectively. All of the above comparisons can verify the effectiveness of our proposed method, which is probably because the training using multilevel noise can retrieve more general degradation information, producing satisfactory denoising performance.

V. CONCLUSION

In this work, we propose an MNC-Net that performs few-shot image denoising with only a few noisy–clean images.

In the first stage, the MNCL learns the rich knowledge of degradation using contrastive learning on the pure noisy images with multilevel synthetic noise levels, which aims to pull in the distance between dataset domains. In the second stage, we utilize limited image pairs to supervise the training of the denoising network consisting of GFE and ADD. Note that our method fuses the contrasting features extracted by CFE with those extracted by GFE and adds our proposed CPCA block to GFE to alleviate the feature domain gap. Extensive experiments proved that the proposed method achieves satisfactory performance, particularly in real-world noisy image denoising. Our work provides a promising direction for image denoising using few-shot learning. In the future, we will focus on proposing few-shot denoising frameworks with more efficiency and better performances.

REFERENCES

[1] D. Zhu *et al.*, "Cascaded normal filtering neural network for geometry-aware mesh denoising of measurement surfaces," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–13, 2021.

[2] B. Zhao, C. Cheng, Z. Peng, Q. He, and G. Meng, "Hybrid pre-training strategy for deep denoising neural networks and its application in machine fault diagnosis," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–11, 2021.

[3] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3142–3155, Jul. 2017.

[4] Z. Yue, H. Yong, Q. Zhao, L. Zhang, and D. Meng, "Variational denoising network: Toward blind noise modeling and removal," 2019, *arXiv:1908.11314*.

[5] S. Anwar and N. Barnes, "Real image denoising with feature attention," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 3155–3164.

[6] H. Shao, D. Zhong, X. Du, S. Du, and R. N. J. Veldhuis, "Few-shot learning for palmprint recognition via meta-siamese network," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–12, 2021.

[7] B. Sun, B. Li, S. Cai, Y. Yuan, and C. Zhang, "FSCE: Few-shot object detection via contrastive proposal encoding," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 7352–7362.

[8] H. Huang, J. Zhang, J. Zhang, J. Xu, and Q. Wu, "Low-rank pairwise alignment bilinear network for few-shot fine-grained image classification," *IEEE Trans. Multimedia*, vol. 23, pp. 1666–1680, 2021.

[9] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. Lillicrap, "One-shot learning with memory-augmented neural networks," 2016, *arXiv:1605.06065*.

[10] G. Koch *et al.*, "Siamese neural networks for one-shot image recognition," in *ICML Deep Learn. Workshop*, vol. 2. Lille, France, 2015, pp. 1–30.

[11] S. Ravi and H. Larochelle, "Optimization as a model for few-shot learning," in *Proc. Int. Conf. Learn. Represent. (ICLR)*. Toulon, France: Palais des Congrès Neptune, 2017, pp. 1–11.

[12] J.-C. Su, S. Maji, and B. Hariharan, "When does self-supervision improve few-shot learning?" in *Proc. Eur. Conf. Comput. Vis.* Scottish Event Campus (SEC), 2020, pp. 645–666.

[13] J. Zhong, X. Bi, Q. Shu, D. Zhang, and X. Li, "An improved wavelet spectrum segmentation algorithm based on spectral kurtogram for denoising partial discharge signals," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–8, 2021.

[14] J. Tang, S. Zhou, and C. Pan, "A denoising algorithm for partial discharge measurement based on the combination of wavelet threshold and total variation theory," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 6, pp. 3428–3441, Jun. 2020.

[15] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-D transform-domain collaborative filtering," *IEEE Trans. Image Process.*, vol. 16, no. 8, pp. 2080–2095, Aug. 2007.

[16] A. Buades, B. Coll, and J.-M. Morel, "A non-local algorithm for image denoising," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 2, Jun. 2005, pp. 60–65.

[17] S. Gu, L. Zhang, W. Zuo, and X. Feng, "Weighted nuclear norm minimization with application to image denoising," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 2862–2869.

[18] X. Mao, C. Shen, and Y.-B. Yang, "Image restoration using very deep convolutional encoder–decoder networks with symmetric skip connections," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 29, 2016, pp. 2802–2810.

[19] Y. Tai, J. Yang, X. Liu, and C. Xu, "MemNet: A persistent memory network for image restoration," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 4539–4547.

[20] K. Zhang, W. Zuo, and L. Zhang, "FFDNet: Toward a fast and flexible solution for CNN-based image denoising," *IEEE Trans. Image Process.*, vol. 27, no. 9, pp. 4608–4622, Sep. 2018.

[21] S. Lefkimmiatis, "Universal denoising networks : A novel CNN architecture for image denoising," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3204–3213.

[22] T. Plötz and S. Roth, "Neural nearest neighbors networks," 2018, *arXiv:1810.12575*.

[23] S. Guo, Z. Yan, K. Zhang, W. Zuo, and L. Zhang, "Toward convolutional blind denoising of real photographs," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 1712–1722.

[24] S. Gu, Y. Li, L. Van Gool, and R. Timofte, "Self-guided network for fast image denoising," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 2511–2520.

[25] J. Lehtinen *et al.*, "Noise2Noise: Learning image restoration without clean data," 2018, *arXiv:1803.04189*.

[26] A. Krull, T.-O. Buchholz, and F. Jug, "Noise2Void—Learning denoising from single noisy images," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2129–2137.

[27] J. Batson and L. Royer, "Noise2Self: Blind denoising by self-supervision," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 524–533.

[28] N. Moran, D. Schmidt, Y. Zhong, and P. Coady, "Noisier2Noise: Learning to denoise from unpaired noisy data," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 12064–12072.

[29] J. Xu *et al.*, "Noisy-as-clean: Learning self-supervised denoising from corrupted image," *IEEE Trans. Image Process.*, vol. 29, pp. 9316–9329, 2020.

[30] Y. Quan, M. Chen, T. Pang, and H. Ji, "Self2Self with dropout: Learning self-supervised denoising from single image," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 1890–1898.

[31] L. Fei-Fei, R. Fergus, and P. Perona, "One-shot learning of object categories," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 4, pp. 594–611, Apr. 2006.

[32] T. Munkhdalai and H. Yu, "Meta networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 2554–2563.

[33] O. Vinyals, C. Blundell, T. Lillicrap, and D. Wierstra, "Matching networks for one shot learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 29, 2016, pp. 3630–3638.

[34] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 1126–1135.

[35] C. Doersch, A. Gupta, and A. A. Efros, "Unsupervised visual representation learning by context prediction," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1422–1430.

[36] R. Zhang, P. Isola, and A. A. Efros, "Colorful image colorization," in *Computer Vision–ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham, Switzerland: Springer, 2016, pp. 649–666.

[37] M. Noroozi, H. Pirsiavash, and P. Favaro, "Representation learning by learning to count," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5898–5906.

[38] S. Gidaris, P. Singh, and N. Komodakis, "Unsupervised representation learning by predicting image rotations," 2018, *arXiv:1803.07728*.

[39] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin, "Unsupervised feature learning via non-parametric instance discrimination," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3733–3742.

[40] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 1597–1607.

[41] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 9729–9738.

[42] Y. Tian, D. Krishnan, and P. Isola, "Contrastive multiview coding," in *Computer Vision—ECCV 2020*. Glasgow, U.K.: Springer, Aug. 2020, pp. 776–794.

[43] A. van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," 2018, *arXiv:1807.03748*.

[44] O. Henaff, "Data-efficient image recognition with contrastive predictive coding," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 4182–4192.

[45] L. Wang *et al.*, "Unsupervised degradation representation learning for blind super-resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 10581–10590.
[46] X. Chen, H. Fan, R. Girshick, and K. He, "Improved baselines with momentum contrastive learning," 2020, *arXiv:2003.04297*.
[47] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7132–7141.
[48] H. Zhao, X. Kong, J. He, Y. Qiao, and C. Dong, "Efficient image super-resolution using pixel attention," in *Proc. Eur. Conf. Comput. Vis.* Scottish Event Campus (SEC), 2020, pp. 56–72.
[49] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Instance normalization: The missing ingredient for fast stylization," 2016, *arXiv:1607.08022*.
[50] E. Agustsson and R. Timofte, "NTIRE 2017 challenge on single image super-resolution: Dataset and study," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 126–135.
[51] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proc. 8th IEEE Int. Conf. Comput. Vis. (ICCV)*, vol. 2, Jul. 2001, pp. 416–423.
[52] A. Abdelhamed, S. Lin, and M. S. Brown, "A high-quality denoising dataset for smartphone cameras," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1692–1700.
[53] S. Roth and M. J. Black, "Fields of experts," *Int. J. Comput. Vis.*, vol. 82, no. 2, p. 205, 2009.
[54] O. Russakovsky *et al.*, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.
[55] T. Plotz and S. Roth, "Benchmarking denoising algorithms with real photographs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1586–1595.
[56] S. W. Zamir *et al.*, "Multi-stage progressive image restoration," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 14821–14831.
[57] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Color image denoising via sparse 3D collaborative filtering with grouping constraint in luminance-chrominance space," in *Proc. IEEE Int. Conf. Image Process.*, vol. 1, Sep. 2007, p. 313.
[58] H. C. Burger, C. J. Schuler, and S. Harmeling, "Image denoising: Can plain neural networks compete with BM3D?" in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 2392–2399.
[59] Y. Chen and T. Pock, "Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1256–1272, Jun. 2016.
[60] K. Zhang, W. Zuo, S. Gu, and L. Zhang, "Learning deep CNN denoiser prior for image restoration," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3929–3938.
[61] S. Lefkimmiatis, "Non-local color image denoising with convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3587–3596.
[62] S. Laine, T. Karras, J. Lehtinen, and T. Aila, "High-quality self-supervised deep image denoising," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 1–11.
[63] P. Singh and R. Shree, "Speckle noise: Modelling and implementation," *Int. J. Control Theory Appl.*, vol. 9, no. 17, pp. 8717–8727, 2016.

Bo Jiang received the B.S. and M.S. degrees in electrification and automation from Northwest Agriculture and Forestry University, Xianyang, China, in 2017 and 2020, respectively. He is currently pursuing the Ph.D. degree in computer applied technology with the Harbin Institute of Technology at Shenzhen, Shenzhen, China.

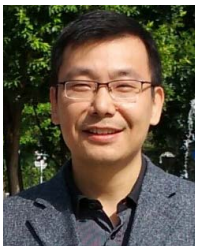
His research interests include deep learning, pattern recognition, computer vision, and low-level image processing.

Jiahuan Wang received the B.S. degree from the Department of Computer Science and Technology, Huazhong Agriculture University, Wuhan, China, in 2019. She is currently pursuing the M.S. degree with the Harbin Institute of Technology at Shenzhen, Shenzhen, China.

Her research interests include image denoising, deep learning, computer vision, and relevant applications.

Yao Lu received the B.S. degree in software engineering from Huaqiao University, Xiamen, China, in 2015, and the Ph.D. degree in computer applied technology from the Harbin Institute of Technology at Shenzhen, Shenzhen, China, in 2020.

She was a Post-Doctoral Fellow with the University of Macau, Macau, China, from 2020 to 2021. She is currently an Assistant Professor with the Biocomputing Research Center, Harbin Institute of Technology at Shenzhen. Her research interests include pattern recognition, deep learning, computer vision, and relevant applications.



Guangming Lu (Senior Member, IEEE) received the B.S. degree in electrical engineering, the M.S. degree in control theory and control engineering, and the Ph.D. degree in computer science and engineering from the Harbin Institute of Technology (HIT), Harbin, China, in 1998, 2000, and 2005, respectively.

He was a Post-Doctoral Fellow with Tsinghua University, Beijing, China, from 2005 to 2007. He is currently a Professor with the Biocomputing Research Center, Harbin Institute of Technology at Shenzhen, Shenzhen, China. He has published over 120 technical papers at prestigious international journals and conferences. His current research interests include pattern recognition, image processing, and automated biometric technologies and applications.

David Zhang (Life Fellow, IEEE) received the M.S. and Ph.D. degrees in computer science from the Harbin Institute of Technology (HIT), Harbin, China, in 1982 and 1985, respectively, and the second Ph.D. degree in electrical and computer engineering from the University of Waterloo, ON, Canada, in 1994.

From 1986 to 1988, he was a Post-Doctoral Fellow with Tsinghua University, Beijing, China, and an Associate Professor with the Academia Sinica, Beijing. From 2005 to 2018, he was a Chair Professor with The Hong Kong Polytechnic University, Hong Kong, where he was the Founding Director of the Biometrics Technology Centre (UGC/CRC) supported by the Hong Kong SAR Government in 1998. He is currently a Presidential Chair Professor with the School of Science and Engineering, The Chinese University of Hong Kong at Shenzhen, Shenzhen, China. He is also a Visiting Chair Professor with Tsinghua University and an Adjunct Professor with Peking University, Beijing; Shanghai Jiao Tong University, Shanghai, China; HIT; and the University of Waterloo. His current research interests include medical biometrics and pattern recognition.